فصل ٦– روش‌های جستجوی یک متغیّره نامقیّد

مقدمه

در این فصل به بررسی سیستم‌های تک‌متغیره می‌پردازیم. ضرورت بررسی این روش‌ها به‌زعمی فهم بهتر، اخذ ایده و سپس تعمیم آنها به سیستم‌های چندمتغیّره می‌باشد. علاوه بر این نکات از این روش‌ها به‌طور مستقیم و مکرّر در برخی تکنیک‌های چندمتغیّره اعمّ از مقیّد و نامقیّد بهره خواهیم‌گرفت.

در یک تقسیم‌بندی کلی می‌توان ایده‌های نشأت‌گرفته را به سه دسته تقسیم‌کرد؛ جستجو براساس حذف بازه‌ها مثل فیبوناچی و تقسیم طلایی، جستجو براساس محاصره و جستجو براساس برازش منحنی. لازم به‌ذکر است که روش‌های مدرن مثل الگوریتم ژنتیک در زمرهٔ این روش‌ها محسوب نمی‌شوند.

روش‌های حذف بازه[۱]

فلسفهٔ اصلی این روش‌ها، برای مسائلی است که باید هزینه زیادی جهت محاسبهٔ تابع هزینه بپردازیم. به‌طور مثال به مهندس کارخانه یا بخش تحقیق و توسعه[۲] می‌گویند دمای بهینه کورهٔ پخت سیمان چقدراست و حق دارد فقط حداکثر ۱۰ آزمایش انجام‌دهد. از طرفی مهندس مزبور هیچ مدل یا شناخت دقیق و کافی از عملکرد کوره ندارد. شاید اولین ایده‌ای که به‌ذهن او برسد، اینست که در بازهٔ عملیاتی کوره (مثلا" ۸۰۰ تا ۱۳۰۰ درجهٔ سانتیگراد) ۱۰ تا تقسیم متساوی‌الفاصله، یعنی هرکدام ۵۰ درجه در نظرگرفته و از فرصت ۱۰ آزمایش استفاده‌کند و دمای بهینه را با تولرانس ۵۰ درجه سانتیگراد، یعنی با وضوح یا عدم‌قطعیت ۵۰ درجه (۲۵± درجه)، دمای اُپتیمم را تعیین کند! حال سئوال اینجاست؛ آیا می‌توان از این فرصت ۱۰ آزمایش، بهرهٔ بیشتری برد و با تولرانس کمتری دمای اُپتیمم را معلوم کرد؟

قبل از پرداختن به پاسخ، نکات و مبادی زیر را طرح می‌کنیم:

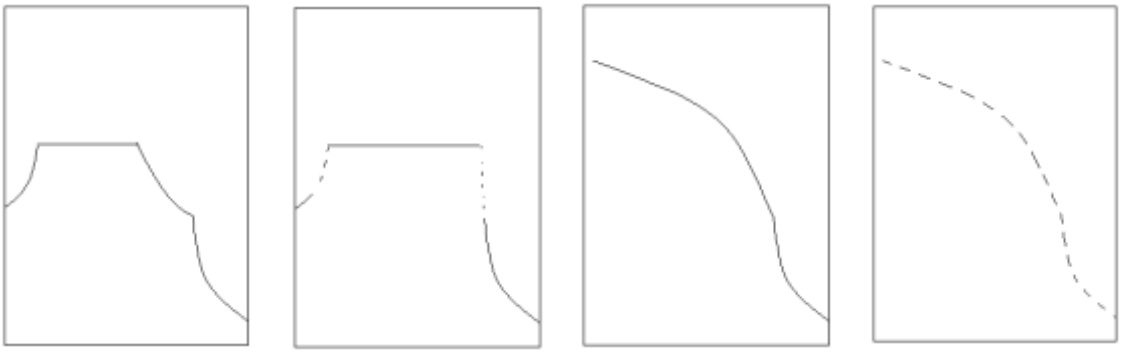
- جستجوی همزمان[۳] در مقابل جستجوی متوالی[۴]: روشی که در بالا گفته‌شد، یک روش جستجوی همزمان است، یعنی از ابتدا نقشهٔ جستجو را معلوم کرده‌ایم در حالیکه ممکنست آزمایشات را طوری ترتیب‌دهیم که آزمایش بعدی پس از معلوم‌شدن نتیجهٔ آزمایش قبلی، طرح‌ریزی شود. در اینصورت می‌گوییم یک جستجوی متوالی خواهیم‌داشت.
- نقشه یا طرح جستجو: ایدهٔ ترتیب‌دادن آزمایشات را می‌گوییم نقشهٔ جستجو[۵]. در حالت یک‌متغیّره نامیده می‌شود، در حالیکه برای چندمتغیره موسوم به طرح یا الگوی جستجو[۶] می‌باشد.
- تخصیص آزمایشات[۷]: عمل یا الفبای نقشه جستجو معروف به تخصیص آزمایشات می‌باشد. به‌طور مثال وظیفه مهندس فوق‌المذکور برای یافتن دمای بهینه، این بود که بگوید محل آزمایشات (چه دماهایی) چه‌باشد. البته طرح او به‌صورت « همزمان» بود.

اصطلاحات بالا عمدتاً با زمینه‌های « مهندسی اطلاعات» و « هوش مصنوعی» هم‌پوشانی داشته و برای جزئیات باید به مراجع و مآخذ مقتضی مراجعه کرد.

---

۱ Region Elimination
۲ Research and development – R&D
۳ Simultaneous Search
۴ Sequential Search
۵ Search Plan
۶ Search Pattern
۷ Experiment Assignment

تعریف - یونیمودال (Unimodal) : یعنی دریک بازه (چه باز و چه بسته) فقط یک اکسترمم (غیر ازمرزها) داشته‌باشیم. تعریف ریاضی را نمی‌نویسیم، برای سادگی وسهولت فقط به شکل‌ها توجه‌کنید:



Continuous Discontinuous Constrained Discrete

نکته: تعریف یونیمودال را می‌توان بدون تعریف یا استفاده ازمشتق انجام داد.

فایده: در بسیاری از مسایل به ویژه فیزیکی و مهندسی گسترهٔ تغییرات متغیر مستقل را میدانیم، پس فعلا با بازه محدود و بسته کار می‌کنیم و فرض می‌کنیم این بازه تردید اولیه یونیمودال است و جواب بهینه واقعاً دراین بین وجود دارد، لذا هدف ما اینست (در مقام پاسخ به سوال اوّل بحث)که فعلاً با یک روش سیستماتیک این بازه تردید را کاهش دهیم. برای سهولت عجالتاً روی محل متغیر مستقل متمرکز هستیم و کاری به تابع هدف نداریم تا حتی‌الامکان همه حالت‌های شکل‌های بالا را بپوشانیم چراکه ممکنست با مشتق تابع و... سروکار داشته باشیم که بحث پیوستگی مطرح می‌شود و قس علیهذا. این از سنتز و هدف طراحی، برای تحلیل چه؟ برای ارزیابی، محک و مقایسه روش‌های حذف بازه یا کاهش دامنه عدم‌قطعیت باید یک شاخص[۸]، ایندکس یا اندازه‌ای از کارآیی داشته‌باشیم. به مثال انگیزشی زیر برای تعریف یک شاخص توجه کنید. (یادآوری می‌شود روش‌هایی که در پیش رو داریم یک متغیره هستند و هدف عالی و فلسفه متعالی بهگزینی، محاسبه یا تعیین اپتیمم با کمترین هزینه و بدون تکرار همه حالات ممکن می‌باشد.)

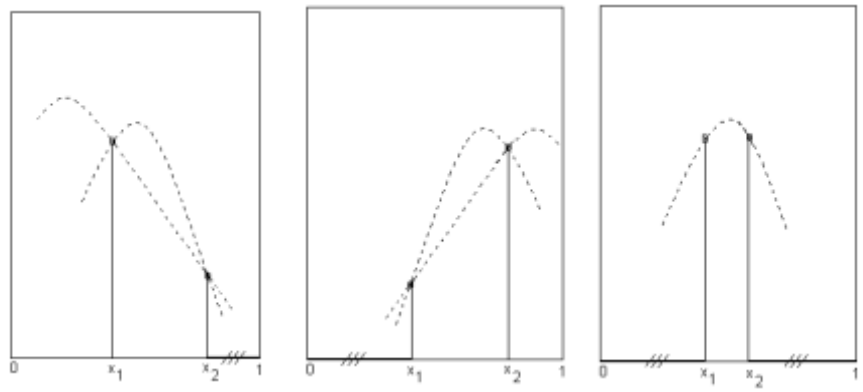
مثال (انگیزشی) ۱- (برای مقدمه‌چینی یک شاخص کارآیی) :

فرض کنید یک تابع هدف یک‌متغیره داریم که مقدار آن در طول گستره متغیر مستقل آن معلوم نیست. اگر حق داشته‌باشیم فقط دو آزمایش انجام دهیم پس از آزمایش، بازه تردید چه‌قدر کوچکتر می‌شود؟ برای سهولت فرض کنید دامنه تغییرات متغیّر مستقل (در اینجا x) از صفر تا یک می‌باشد. به شکل صفحه بعد رجوع کنید، کلاً سه حالت ممکنست پیش بیاید. در حالت اول چون مقدار y_1 از y_2 بزرگتر شده‌است باید مقدار بهینه (x^*) متغیّر مستقل (متغیّر تصمیم‌گیری) بین مقدار صفر و x_2 قراربگیرد یعنی مطمئن هستیم که x^* در فاصله x_2 تا 1 قرار ندارد. در حالت دوم به طریق مشابه به حذف یک قسمت از سه قسمت گستره x خواهیم رسید. حالت سوم نیز بسیار بعید است. پس به‌طور آماری می‌توان گفت: همیشه یک قسمت از سه قسمت حذف می‌شود. بدین ترتیب اگر search plan ما همین‌طور باشد ولی محل دو آزمایش به‌طور متساوی‌الفاصله در بازه اولیه انتخاب شود، آنگاه سرعت کاهش بازه تردید میشود $(2/3)^k$ بطوریکه k تعداد آزمایشات است. به این نقشه جستجو می‌گویند نقشه جستجوی متساوی‌الفاصله دونقطه‌ای[۹].

[۸] measure

[۹] Two-point Equal Interval Search

منحنی‌های نقطه‌چین، رفتار تابع بطور فرضی می‌باشد و علامت $\#\#$ به‌معنی حذف بازه میباشد.



$y_1 = y_2$	$y_1 \prec y_2$	$y_1 \succ y_2$
$x_1 \le x^* \prec x_2$	$x_1 \le x^* \prec 1$	$0 \le x^* \prec x_2$
(حالت سوم)	(حالت دوم)	(حالت اول)

برای مثال خودمان

$$L^k = (2/3)^k L^\circ \quad , \quad L^\circ = [0,1]$$

به‌شرطیکه : $x_1 = 0 = 1 - x_2 = x_2 - x_1$

اگر بگوییم x_1 و x_2 آنقدر به هم نزدیک انتخاب شوند که تقریباً در حالت حدی، نصف بازه حذف شود، آنگاه به روش تنصیف[۱۰] رسیده‌ایم، با سرعت کاهش $(1/2)^k$:

$$\begin{cases} L^k \cong (1/2)^k L^\circ \\ x_1 - 0 \cong 1 - x_2 \end{cases}$$

برای $k = 8$ امتحان کنید

روش اول : $L^8 = (2/3)^8 L^\circ \cong 4\% L^\circ$

روش دوم : $L^8 = (1/2)^8 L^\circ \cong 0.4\% L^\circ$

بدین‌ترتیب، با تغییر محل آزمایش، سرعت کاهش بازه تغییر می‌کند. به‌هرحال این‌ها استشمامی و حسّی هستند و می‌خواهیم بدانیم بهترین نقشه جستجو چیست؟

طراحی میزان کارآیی جستجو (نوعاً جستجوی همزمان)

نکته ۱: میزان مربوطه باید مستقل از نوع و ساختار تابع هدف باشد.

نکته ۲: میزان مربوطه باید مستقل از شانس و افزودگی[۱۱] باشد.

نکته ۳: میزان مربوطه باید تابع محلّ آزمایشات (x_i ها) ولی مستقل از خروجی آزمایش باشد.

یک میزان پیشنهادی خوب که سه خاصیت بالا را داشته باشد عبارتست از:

سایز (اندازه) بزرگترین بازه تردید بعد از تعیین محل آزمایشات.

[۱۰] Bisection or Dichotonomous

[۱۱] Bias

برای مأنوس‌شدن ایده به شکل نگاه کنید، مسأله یک مسأله ماگزیمم‌سازی است و قرار است سه آزمایش اجرا شود. دو نقشه جستجو داریم، یکی غیرمتساوی‌الفاصله و دیگری متساوی‌الفاصله:

نقشه جستجوی I: سه آزمایش در 10%، 40%و 80% بازه اولیه.

نقشه جستجوی II: سه آزمایش در 25%، 50% و 75% بازه اولیه.

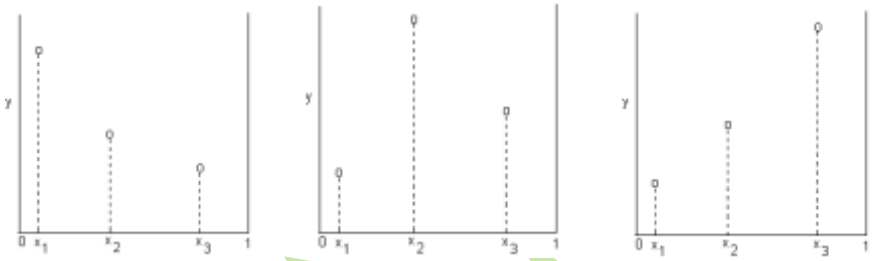
(یادآوری: ۱ + تعداد نقاط آزمایشات = تعداد تقسیمات)

Search Plan I:

$x_1 = 0.1,$

$x_2 = 0.4,$

$x_3 = 0.8$

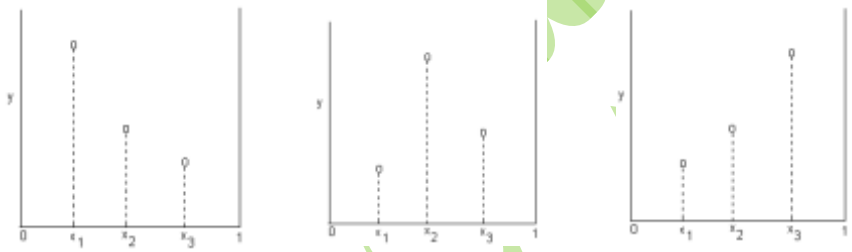


Search Plan II:

$x_1 = 0.25,$

$x_2 = 0.5,$

$x_3 = 0.75$



تعریف K: ایندکس بزرگترین مقدار تابع یا بهره تابع[۱۲]

برای جستجوی شماره I محلّ بازه تردید نهایی بستگی به مقدار تابع آزمایش دارد ولی فعلاً دنبال سایز یا اندازه هستیم.

برای جستجوی I :

خوش‌شانس‌ترین	$i_3 = 0.4 - 0.0$	$0 \prec x^* \prec x_2$	آنگاه	$K = 1$	اگر
بدشانس‌ترین	$i_3 = 0.8 - 0.1$	$x_2 \prec x^* \prec x_3$	آنگاه	$K = 2$	اگر
متوسّط‌الاقبال	$i_3 = 1.0 - 0.4$	$x_2 \prec x^* \prec 1$	آنگاه	$K = 3$	اگر

نتیجه بالا را به‌طور خلاصه می‌نویسم، به‌طوریکه i_3 تعریف می‌شود به اندازه بازه تردید بعد از 3آزمایش.

$$i_3 = [0.4, 0.7, 0.6]$$

برای جستجوی II : به‌طریق مشابه $i_3 = [0.5, 0.5, 0.5]$

حال اگر میزان را بزرگترین بازه تردید تعریف کنیم:

$$\text{Search Plan I}: \quad I_3 = \max_{1 \le K \le 3}[I_3] = \max[0.4, 0.7, 0.6] = 0.7$$

$$\text{Search Plan II}: \quad I_3 = \max_{1 \le K \le 3}[I_3] = \max[0.5, 0.5, 0.5] = 0.5$$

پس نقشه II، بهتراست از نقشه I.

[۱۲] Outcome

حال اگر تعمیم دهیم برای n آزمایش (برای سهولت، شکل را نمی‌کشیم و فرض یونیمودال را فراموش نمی‌کنیم)

$$i_n = [x_{k+1} - x_{k-1}]$$

بدین ترتیب یک معیار محک و مقایسه جستجوها می‌شود: $I_n(\underline{X}_n) = \max[i_n(\underline{X}_n, K)]$

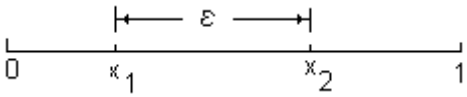
اصل مینی‌ماکس[۱۳]

عبارت بالا و شعار ما برای تعریف محک، همان اصل مینی‌ماکس می‌باشد.

$$I_n^* = I_n(\underline{X}_n^*) = \min[I_n(\underline{X}_n)]$$

به عبارت ساده‌تر، یعنی روی هرچه نقشه جستجو که با n آزمایش کار می‌کند، جستجو، مقایسه و بررسی کنیم، آن نقشه‌ای بهترین است که ماگزیمم بازه تردید نهایی محتمل کوچکتری بدهد.

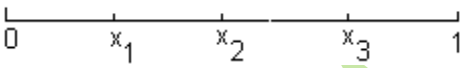
مثال۲- اصل مینی ماکس را برای $n = 2$ و $n = 3$ به‌دست آورید.



$$I_2 = \max[x_2, 1 - x_1] \rightarrow I_2^* = \min I_2$$

این عبارت موقعی مینیمم می‌شود که $x_1 = x_2$ باشد، در عمل این ممکن نیست مگر با مقداری اختلاف[۱۴] موسوم به ε.

$$x_1 = x_2 \rightarrow \text{Bisecting جستجوی} \rightarrow I_2^* = 0.5 + \varepsilon/2 \quad \text{فرض}$$



$$I_3 = \max[x_2, x_3 - x_1, 1 - x_2] \rightarrow I_3^* = \min I_3$$

$$\rightarrow I_3^* = 0.5 \quad \text{اگر متساوی‌الفاصله باشند، } I_3 \text{ مینیمم می‌شود}$$

.

خلاصه و نتیجه:

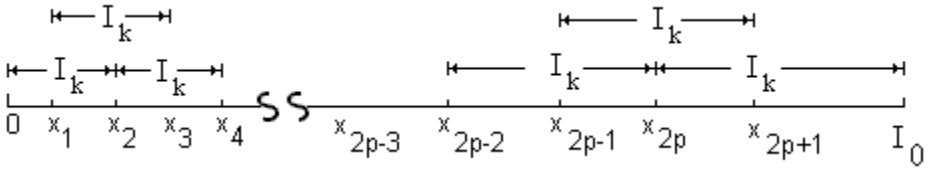
اصل مینیماکس برای $n = 2$ $I_2^* = 0.5 + \varepsilon/2$

اصل مینیماکس برای $n = 3$ اضافه کردن یک آزمایش, خیلی تأثیر ندارد. $I_3^* = 0.5$

روش‌های جستجوی مستقیم همزمان (در مقابل متوالی)

حال می‌خواهیم با استفاده از اصل مینی‌ماکس به محاسبه I_K^* بپردازیم، یعنی به‌جای 2 یا 3 آزمایش، اگر K آزمایش داشته‌باشیم، میزان و معیار کارآئی چه خواهد بود. جهت سهولت، تعداد آزمایشات را به فرد و زوج تقسیم می‌کنیم. فرض کنید K فرد است، یعنی $K = 2p+1$ و بازه تردید اولیه $I_\circ$ است. همان‌طور که از مثال‌های اخیر دیدیم بازه‌های تردید با دوتا فاصله اختلاف قرار می‌گیرند، یعنی:

$$(x_2 - 0), (x_3 - x_1), (x_4 - x_2), \cdots, (x_{2p} - x_{2p-2}), (x_{2p+1} - x_{2p-1}), (I_\circ - x_{2p})$$



به‌خاطر تعریف I_K باید این روابط برقرار باشد:

---

[۱۳] MiniMax

[۱۴] Resolution

$$+\begin{cases} x_2 - 0 \le I_K \\ x_4 - x_2 \le I_K \\ \vdots \\ x_{2p} - x_{2p-2} \le I_K \\ I_\circ - x_{2p} \le I_K \end{cases}$$

$$I_\circ \le (p+1) I_k \quad \rightarrow \quad I_K \ge \frac{I_\circ}{p+1} \quad \rightarrow \quad I_K^* \ge \frac{I_\circ}{p+1}$$

حال اگر محل آزمایشات زوج مثل x_2 و x_4 و ... باشد، بدیهیست که متساوی‌الفاصله باشند، یعنی:

$$x_{2h} = hI_K^* \quad , \quad h = 1,2,\cdots,p$$

ولی برای محل آزمایشات فرد (ایندکسهای فرد):

یک راه اینست که فردها را در هر فاصله‌ای از زوجها که دوست داشته‌باشیم بگذاریم، مثلا x_1 را در 40 درصدی x_0 و x_2 بگذاریم و x_3 را بین x_2 و x_4 (یکجایی این‌بین). لذا یک جستجوی یگانه برای آزمایشات با ایندکس فرد نداریم. معذلک خودمان قرار می‌گذاریم در بین زوجها، در وسط قرار دهیم: (بر حسب بازه نهایی اگر بنویسیم)

$$x_j = jI_K^*/2 \quad , \quad j = 1,2,\cdots,K \qquad \text{یا}$$

$$x_j = jI_\circ / 2(p+1) \quad , \quad j = 1,2,\cdots,K \qquad \text{(بر حسب بازه اولیه)}$$

این روند موسوم به جستجوی یکنواخت است[۱۵].

حال فرض کنید تعداد آزمایشات زوج است، در این‌صورت یک جستجوی یگانه داریم.

$$+\begin{cases} x_2 - 0 \le I_{2p} \\ x_4 - x_2 \le I_{2p} \\ \vdots \\ x_{2p} - x_{2p-2} \le I_{2p} \end{cases}$$

$$x_{2p} \le pI_{2p} \quad \rightarrow \quad \text{(باید به‌نحوی } I_\circ \text{ را دخیل کنیم)} \quad \rightarrow$$

اینتروال آخر شامل $I_\circ$ و یک نقطه فرد است:

$$I_\circ - x_{2p-1} \le I_2$$

روابط اخیر را جمع میکنیم:

$$I_\circ + x_{2p} - x_{2p-1} \le (p+1) I_{2p}$$

برای محاسبه I_{2p}^* (یعنی مینیمم‌کردن I_{2p}) باید علامت مساوی را انتخاب‌کنیم و فاصله دو اینتروال $x_{2p} - x_{2p-1}$ نیز باید بسیار کم باشد، یعنی به اندازه همان اختلاف ε:

$$Resolution = \varepsilon I_\circ \qquad \text{(یعنی کسری از بازه اولیه)}$$

با جایگذاری داریم:

$$I_\circ + \varepsilon I_\circ = (p+1) I_{2p}^* \quad \rightarrow \quad I_{2p}^* = \frac{(1+\varepsilon) I_\circ}{p+1}$$

محل آزمایشات: آزمایشات زوج که معلوم است متساوی‌الفاصله هستند، یعنی:

$$x_{2h} = \frac{h(1+\varepsilon) I_\circ}{p+1} \quad , \quad h = 1,2,\cdots,p$$

و آزمایشات فرد، باید سمت چپ آزمایشات زوج به فاصله اختلاف (ε) مسأله باشند:

$$x_{2h-1} = x_{2h} - \varepsilon I_\circ \;,\; h = 1,2,\cdots,p \rightarrow \text{از بالا } x_{2h} \text{ جاگذاری} \rightarrow x_{2h-1} = \frac{[h-(p+1-h)\varepsilon] I_\circ}{p+1} \;,\; h = 1,2,\cdots,p$$

پس به‌طور خلاصه:

محل آزمایشات: یگانه	$I_{2p}^* = I_\circ / (p+1) + \varepsilon I_\circ / (p+1)$	زوج
محل آزمایشات: یکنواخت	$I_{2p+1}^* = I_\circ / (p+1)$	فرد

[۱۵] Uniform Search

به میزان بهبود زوج و فرد توجه کنید، به اندازه $\varepsilon I_\circ / (p+1)$ که عدد کوچکی می‌باشد، انگار که اصلاً مستقل از فرد و زوج بودن، همان یونیفورم بگیریم بهتر است (از نظر ریاضی $p \to \infty$)، یعنی همان حدس و حس اولیه خودمان، طبیعی (ولی نه الزاماً عقلانی)، همانست که در مثال اول بحث گفتیم (کوره سیمان، تعیین آزمایشات 50 درجه - 50 درجه).

مثال۳- تابع زیر را به عنوان نمونه[۱۶] در نظر بگیرید. مطلوبست محاسبه ماگزیمم با چهار آزمایش و پنج آزمایش، $I_\circ = 1$ ، $\varepsilon = 0.05$ ، نتایج را (طول بازه تردید و محل آزمایشات) با مقدار حقیقی (از روی مشتق) مقایسه کنید:

$$Maximize: \quad y = 3 + 6x - 4x^2 \quad on \quad 0.0 \le x \le 1.0$$

برای چهار آزمایش (غیر متوالی) :

$$x_{2h} = h(1+\varepsilon)I_\circ/(p+1) \quad , \quad h = 1,2$$

$h=1 \rightarrow$
$$\begin{cases} x_2 = (1+\varepsilon)I_\circ/(p+1) = (1+0.05)(1)/(2+1) = 0.35 \\ x_1 = [1-(2+1-1)(0.05)](1)/(2+1) = 0.3 \end{cases}$$
(فرد آن، باید سمت چپ x_2 بهاندازه $\varepsilon I_\circ$ باشد)

$h=2 \rightarrow$
$$\begin{cases} x_4 = 0.70 \\ x_3 = 0.7 - 0.05 = 0.65 \end{cases}$$

محاسبه تابع: آزمایشات $0.3, 0.35, 0.65, 0.7$

$$\begin{cases} y(0.30) = 4.44 \\ y(0.35) = 4.60 \\ y(0.65) = 5.21 \\ y(0.70) = 5.24 \end{cases} \quad (\text{چهار آزمایش}) \qquad I_4^* = 0.35 \qquad 0.65 \le x^* \le 1.0$$

برای پنج آزمایش (غیر متوالی):

$$x_1 = I_\circ/(p+1)2 = 1/2(2+1) = 1/6 \quad for \quad j = 1$$

به‌طریق مشابه:

$$\begin{cases} y(1/6) = 3.89 \\ y(1/3) = 4.56 \\ y(1/2) = 5.00 \\ y(2/3) = 5.22 \\ y(5/6) = 5.22 \end{cases} \quad (\text{پنج آزمایش}) \qquad \begin{aligned} x_2 &= 1/3 \\ x_3 &= 1/2 \\ x_4 &= 2/3 \\ x_5 &= 5/6 \end{aligned}$$

$$I_5^* = 1/3 = 0.333 \quad \rightarrow \quad 2/3 \le x^* \le 1.0$$

حالا مقایسه کنید: تئوری کلاسیک : $x^* = 3/4 \quad , \quad y(x^*) = 5.25$

توجه: بحث اخیر برای دستگرمی، برقراری تعاریف ریاضی و نمادگذاری بود تا مقدمه‌ای باشد برای جستجوهای متوالی. برای جزئیات بیشتر مثل مفهوم resolution , بحث Distinguishability (تمیز دادن) و Scaling باید به کتب و مراجع مناسب رجوع کرد.

روشهای جستجوی متوالی[۱۷]

منظور از متوالی این است که از اطلاعات خروجی آزمایش(مقدار تابع) استفاده کرده و محل آزمایش بعدی را تعیین کنیم. اگر همان مثال کوره سیمان را بگوییم باید آزمایشات مختلف را یکدفعه اعلام نکنیم، بلکه دانه به دانه بگوئیم. خواهید دید بازه تردید نهایی بسیار کوچک می‌شود.

یکی از قوی‌ترین طرح‌های جستجو، روش جستجوی فیبوناچی[۱۸] می باشد. در روش مینیماکس باید مقدار آزمایشات از قبل معلوم باشد، در حالی‌که روش فیبوناچی به‌طور متوالی عمل میکند. روش دیگری داریم به‌نام جستجوی تقسیم طلایی[۱۹] که تقریبا با فیبوناچی برابری می‌کند ولی نیازی به تعیین آزمایشات از قبل ندارد. روش دیگری هست که برای توابع گسسته[۲۰] کار می‌کند، موسوم به جستجوی شبکه[۲۱] که در حوصله درس ما نیست.

---

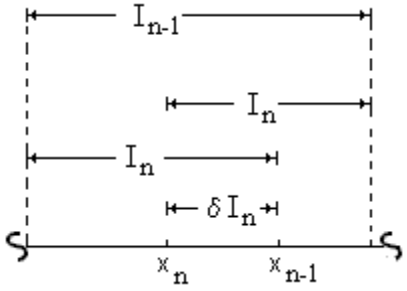
[۱۶] Benchmark
[۱۷] Sequential Search Methods
[۱۸] Fibonacci
[۱۹] Golden Section Method
[۲۰] Discrete
[۲۱] Lattice Search

همچنین یک کلاس از روش جستجوهای مستقیم یکمتغیره وجود دارد که ترکیبی از روش های غیرمتوالی و متوالی میباشند، بنام‌های Even block ، Odd block و Golden block که علی‌رغم کاربردشان در اسکوپ و حوصله درس ما نیست، به‌ویژه اثبات و تحلیل مینی‌ماکس آنها. به کتب مرجع مراجعه شود.

روش جستجوی فیبوناچی

مبدعین آن، Beightler و Wilde می‌باشند و اثبات روش، ریشه در برنامه‌ریزی پویا[۲۲] دارد. برای مسأله ماگزیمم‌سازی به‌دست‌آوردن روابط دست اندرکار به‌طور خلاصه به‌شرح زیر است:

نکته روش در این است که دو آزمایش مانده به آخر را در بازه نهایی به روش غیر متوالی تعیین می‌کنیم؛ سپس از همان آخر می‌آئیم عقب تا به دو آزمایش اول برسیم. به شکل توجه کنید، همانطور که معلومست، محلّ دو آزمایش آخر نسبت به مرکز بازه قبلی (I_{n-1}) قرینه هستند، به‌همین خاطر از هر دو طرف اگر اندازه بگیریم می‌شود I_n و نکته مهمّ و ابداعی این است که دو آزمایش آخر به فاصله δI_n (یا همان $\varepsilon I_\circ$) از هم جدا شده‌اند و این شرط اساسی مسئله است. اگر از این کمتر باشد، تمام محاسبات به‌هم می‌خورد و اگر هم بیشتر باشد، دیگر فیکس کردن تعداد آزمایشات بی‌معنی است.



نکته: اینکه باید قرینه باشند از نظر تئوری برنامه‌ریزی پویا بدست می‌آید که معرف بهترین نقشه جستجو میباشد.

نکته: نکته جالب این روش متوالی این است که سایز بازه‌های تردید قابل طراحی هستند ولی محل آزمایشات این‌طور نیست، بلکه بستگی به حاصل آزمایش‌ها دارد، یعنی باید y های متناظر باهم مقایسه شوند تا آزمایش بعدی (یا قبلی !) به‌دست آید.

طراحی سایز I_n: چون محل آزمایشات برای I_n ، دوتائی همزمان است، از روابط اخیرمان برای $n=2$ استفاده می کنیم:

$$I_2^* = 0.5 + \varepsilon/2$$
$$I_n = 0.5(I_{n-1}) + \delta I_n/2 \quad \rightarrow \quad I_n = I_{n-1}/(2-\delta)$$

همچنین از اصل قرینه‌بودن می‌توان ثابت‌کرد که: $I_{n-2} = I_n + I_{n-1}$

با ترکیب معادلات اخیر و تکرار آن برای ایندکس‌های متفاوت:

$$I_{n-2} = I_n + (2-\delta)I_n = (3-\delta)I_n$$
$$I_{n-3} = (5-2\delta)I_n$$
$$I_{n-4} = (13-5\delta)I_n$$
$$\vdots$$
$$I_{n-j} = (A_{j+2} - A_j\delta)I_n$$

رابطه برگشتی فیبوناچی[۲۳]:

A_j همان دنباله معروف فیبوناچی است، برای تولید آن در کامپیوتر از فرمول زیر کمک بگیرید:

$$A_0 = 0$$
$$A_1 = 1$$
$$A_n = A_{n-1} + A_{n-2}$$ (جمله بعدی جمع دو جمله قبلی است.)

برای $n=12$ در جدول زیر A_j ها آمده است:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
A_n	0	1	1	2	3	5	8	13	21	34	55	89	144

بنابر این می‌توان طول بازه تردید نهایی را بدست آورد:

$$n-j=1 \quad \rightarrow \quad \begin{cases} I_1 = (A_{n+1} - A_{n-1}\delta)I_n \\ \varepsilon I_\circ = \delta I_n \quad , \quad I_1 = I_\circ \end{cases}$$

$\rightarrow$ چون دو آزمایش برای کاهش بازه تردید داریم

رابطه تجمیعی[۲۴] فیبوناچی: $I_n = [(1+\varepsilon A_{n-1})/A_{n+1}]I_\circ$

---

[۲۲] Dynamic Programming

[۲۳] Recursive

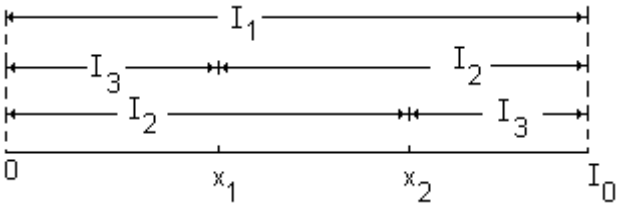
[۲۴] Cumulative

استفاده از رابطه برگشتی: می توان بازه دوم را به‌دست‌آورد:

$$n-j=2 \;\rightarrow\; I_2=(A_n-A_{n-2}\delta)I_n \;\rightarrow\; (\varepsilon \text{ برحسب}) \qquad I_2=A_nI_n-\varepsilon A_{n-2}I_\circ$$

استفاده از رابطه تجمیعی: با داشتن بازه اولیه ($I_\circ$)، تعداد آزمایشات (n) و وضوح (ε) می‌توان طول بازه تردید نهایی (تولرانس) به‌دست آورد.

دقّت شود به‌خاطر تقارن، I_2 همان x_2 نیز هست، به‌شرطی‌که سمت چپ صفر باشد و در حالت کلی، x_2 به‌اندازه I_2 از سمت چپ فاصله دارد.



محل x_1 را نیز بخاطر تقارن با x_2 می‌توان بدست‌آورد و از روی فرمول هم میتوان محاسبه‌کرد:

$$n-j=2 \;\rightarrow\; x_1=I_3=(A_{n-1}-A_{n-3}\delta)I_n=A_{n-1}I_n-\varepsilon A_{n-3}I_\circ$$

تا اینجا بحث تحلیلی داشتیم، یعنی معرفی روش فیبوناچی که تا چه‌قدر میتواند عدم‌قطعیت را کاهش‌دهد.

الگوریتم: بعد از انتخاب دو نقطه اول(x_1 و x_2) در بازه اولیه، باید محل x_3 را بگوئیم که قطعا نسبت به قبلی خود قرینه باشد و همین‌طور مسأله ادامه پیدا می‌کند. فقط خوبی روش در این است که پس از معلوم شدن n و ε می‌توان فهمید تا چه قدر ریز خواهیم‌شد.

مثال۴- همان مثال مجذوری قبلی را با فیبوناچی حل کرده و مقایسه کنید.

$$Maximize: \quad y=3+6x-4x^2 \quad , \quad I_\circ=1.0\,,\ \varepsilon=0.05\,,\ n=4$$

حل: از رابطه تجمیعی می‌توان بازه تردید نهایی را به‌دست‌آورد.

$$I_n=[(1+\varepsilon A_n-1)/A_{n+1}]I_\circ \;\rightarrow\; I_4=[(1+(0.05)(2)/5](1)=0.22$$

محلّ دومین آزمایش:

$$x_2=I_2=A_nI_n-\varepsilon A_{n-2}I_\circ \;\rightarrow\; x_2=3(0.22)-(0.05)(1)(1)=0.61$$

محلّ اولین آزمایش (قرینه نسبت به x_2 : $1-0.61$، ولی از روی رابطه):

$$x_1=I_3=(A_{n-1}I_n-A_{n-3}\delta)I_n=A_{n-1}I_n-\varepsilon A_{n-3}I_\circ=1.0-0.61=0.39$$

ادامه (شروع روش های تکرار)، باید مقادیر تابع را به‌دست آوریم تا بگوئیم x_3 کجاست:

$$y(x_1=0.39)=4.732 \quad , \quad y(x_2=0.61)=5.172$$

پس $0.39\le x^*\le 1.0$، x_3 نسبت به x_2 در این بازه به‌دست می‌آید:

$$x_3=0.39+(1-0.61)=0.78 \;\rightarrow\; y(0.78)=5.246$$

پس $0.61\le x^*\le 1.0$، x_4 نسبت به x_3 در این بازه به‌دست میآید:

$$x_4=0.61+(1-0.78)=0.83 \;\rightarrow\; y(0.83)=5.244$$

مقایسه کنید: مقدار واقعی $y^*(3/4)=5.25$ و دو محل x_3 و x_4 .

خلاصه: روش فیبوناچی سریع‌ترین کاهش بازه تردید را دارد ولی عیب آن این است که باید تعداد آزمایشات معلوم باشد. برای پی بردن به سرعت کاهش بازه تردید، کافیست فرمول تجمیعی گفته شده را در حالت حدی $\varepsilon\to 0$ به‌دست آوریم:

$$I_n=\frac{I_\circ}{A_{n+1}}$$

یعنی اگر 11 آزمایش انجام دهیم، بازه نهایی 144 برابر کوچک می‌شود، پس با آن ایده اول که قرار بود 10 یا 11 قسمت مساوی تقسیم‌کنیم (جستجوی متساوی‌الفاصله)، چه‌قدر تفاوت دارد، 1/11 کجا و 1/144 کجا.

روش جستجوی متوالی تقسیم طلایی[۲۵]

این روش تعمیم و بهبود فیبوناچی می‌باشد. علت بهبود، حذف الزام تعداد آزمایشات معلوم می‌باشد و این کار را به بهای فداکردن جستجوی مینی‌ماکس انجام می‌دهد. این روش از نظر سرعت کاهش تا $n=6$ شبیه فیبوناچی است وتقریبا همان کارائی را با n بزرگ از خود بروز میدهد. نکته آن به یک اصل باستانی، یعنی تقسیم یک خط به نسبت $m:n$ برمیگردد. روش شروع مثل فیبوناچی است، یعنی $x_1=0.382$ و $x_2=0.618$ (منظور 38.2 درصدی و 61.8 درصدی فاصله اولیه می‌باشد. و نکته اصلی آن نگهداشت نسبت بازه های تردید است، برای این کار از رابطه برگشتی فیبوناچی استفاده می‌کنیم:

$$I_{n-j}=I_{n-(j-1)}+I_{n-(j-2)} \rightarrow \quad I_{n-(j-2)} \text{ طرفین تقسیم بر} \rightarrow \quad \frac{I_{n-j}}{I_{n-(j-2)}}=\frac{I_{n-(j-1)}}{I_{n-(j-2)}}+1$$

حال نسبت بازه‌های متوالی را به شکل τ تعریف می‌کنیم:

$$\tau \triangleq \frac{I_{n-j}}{I_{n-(j-1)}}=\frac{I_{n-(j-1)}}{I_{n-(j-2)}}=\cdots=\frac{I_1}{I_2} \rightarrow \text{ جایگزینی در رابطه بالا} \rightarrow \frac{I_{n-j}}{I_{n-(j-2)}}=\frac{I_{n-j}}{I_{n-(j-1)}}\times\frac{I_{n-(j-1)}}{I_{n-(j-2)}}=\tau\times\tau=\tau^2$$

اگر این رابطه را با همان رابطه برگشتی ترکیب‌کنیم:

$$\tau^2=\tau+1 \quad \rightarrow \quad \tau=(1+\sqrt{5})/2=1.618033989...$$

پس برای شروع جستجو، بعد از اینکه اولین یا دومین فیکس شد، از نسبت مربوطه برای بازه بعدی استفاده‌کرده و با استفاده از مقادیر تابع، محل آن را تعیین میکنیم:

$$x_2=I_2=I_1/\tau=I_\circ/\tau=0.618I_\circ$$

و بعد از تعیین قرینه،

$$x_1=I_\circ-I_\circ/2=(1-1/\tau)I_\circ=0.382I_\circ$$

با حفظ نسبت τ، مسأله را ادامه می‌دهیم، جالب است که با روش فیبوناچی مقایسه شود:

$$I_n=\frac{I_\circ}{\tau^{n-1}}$$

مثال۵- همان مثال قبلی

$$Maximize: \quad y=3+6x-4x^2 \quad , \quad I_\circ=1.0$$

بازه نهایی تردید اگر بخواهیم تا 4 بار آزمایش کار کنیم:

$$I_4=1/(1.618)^3=0.236$$

$$\begin{cases} x_2=I_\circ/\tau=1/\tau=0.618 \rightarrow & y(0.618)=5.18 \\ x_1=1-0.618=0.382 \rightarrow & y(0.382)=4.71 \end{cases} \rightarrow \quad 0.382\le x^*\prec 1$$

x_3 قرینه نسبت به x_2 : $x_3=0.382+(1-0.618)=0.764 \rightarrow y(0.764)=5.24 \rightarrow 0.618\le x^*\prec 1.0$

x_4 قرینه نسبت به x_3 : $x_4=0.618+(1-0.764)=0.854 \rightarrow y(0.854)=5.2 \rightarrow 0.618\le x^*\prec 0.854$

خلاصه این دو روش پرمصرف:

- تا زمانی که کار تحلیلی نخواهیم، می‌توان الگوریتم فیبوناچی و تقسیم طلایی را به‌راحتی نوشت.
- برای نوشتن الگوریتم، برای فیبوناچی تعداد آزمایشات مهم است ولی برای تقسیم طلایی فقط به‌درد حد بالای لوپ تکرار می‌خورد.
- برای فیبوناچی در هر تکرار[۲۶] چهار حالت باید در نظرگرفت ولی برای تقسیم طلایی، دو حالت.

---

[۲۵] Golden Section

[۲۶] Iteration

روش‌های جستجو و محاصره[۲۷]

کلاس دیگری از روش‌های پایه جستجوی یک‌متغیره، روش‌هایی تکراری هستند که بر مبنای سعی و خطا استوارند. یک شاخه از این فامیلی، بر مبنای محاصره کار می‌کنند. علت وجودی این روش‌ها، تضمین همگرایی و مشکلات روش‌های هندسی- تکراری در حالت بازه تردید اولیه باز (چه یکسره و چه دوسره) می‌باشد.

مبنای این روش‌ها متکی به دو مشخصه و مؤلفه مهم است. یکی، حدس(های)اولیه ودیگری موتور تکرار یا تعیین خودکارحدس بعدی با استفاده از اطلاعات قبلی متغیرمستقل ومتغیرتابع. ازاین‌رو، این روش‌ها موسوم به منطقی[۲۸] ومتوالی (تکراری[۲۹]) منطقی هستند، چون براساس منطق بهبود جواب درهرمرحله یا گام ازسعی وخطا هستند، چون با سعی وخطا کار می‌کنند. به خاطرماهیت سعی وخطای این روش‌ها، محکّ ومقایسه آنها معمولاً با سرعت همگرایی[۳۰] سنجیده می‌شود.

سرعت همگرایی خطی-تعریف این میزان، به‌صورت نسبت فاصله دوحدس متوالی تا جواب نهایی می‌باشد:

برای یک‌متغیره

$$\frac{\left|x^{k+1}-x^{*}\right|}{\left|x^{k}-x^{*}\right|}\le c \quad , \quad 0\le c\le 1$$

برای چندمتغیره

$$\frac{\left\|\underline{x}^{k+1}-\underline{x}^{*}\right\|}{\left\|\underline{x}^{k}-\underline{x}^{*}\right\|}\le c$$

سرعت همگرایی درجه p ام– این تعریف ، عملا تعریف کلی از نسبت اخیرالذکر بر مبنای نورم درجه p می‌باشد:

$$\frac{\left\|\underline{x}^{k+1}-\underline{x}^{*}\right\|^{p}}{\left\|\underline{x}^{k}-\underline{x}^{*}\right\|^{p}}\le c \quad , \quad c\ge 0 \quad , \quad p\ge 1$$

هرچه p بزرگتر باشد، به‌این معنی است که سرعت همگرایی بیشتر است. به‌طور مثال اگر $p=1$ باشد، به همان تعریف سرعت همگرایی خطی می‌رسیم که معمولا کند اطلاق می‌شود، در حالی‌که اگر $p=2$ باشد، اطلاق به سرعت همگرایی درجه دوم شده و از سرعت خطّی تندتر است.

سرعت همگرایی ابرخطّی[۳۱] – این تعریف ، در خود مفهوم مجانبی دارد:

$$\lim_{k\to\infty}\frac{\left\|\underline{x}^{k+1}-\underline{x}^{*}\right\|^{p}}{\left\|\underline{x}^{k}-\underline{x}^{*}\right\|^{p}}\to 0$$

نکته – روش‌هایی که در ادامه می‌آیند، موسوم به روش‌های غیرمستقیم[۳۲] در مقابل روش‌های جستجوی مستقیم نظیر تقسیم طلایی و فیبوناچی هستند. علّت آن نیز معلوم است، به‌جای جستجوی مستقیم مینیمم (ماگزیمم) به‌طور غیرمستقیم سعی در یافتن ریشه مشتق تابع هدف ($f'(x)=0$ ، شرط لازم اکسترمم) دارند.

---

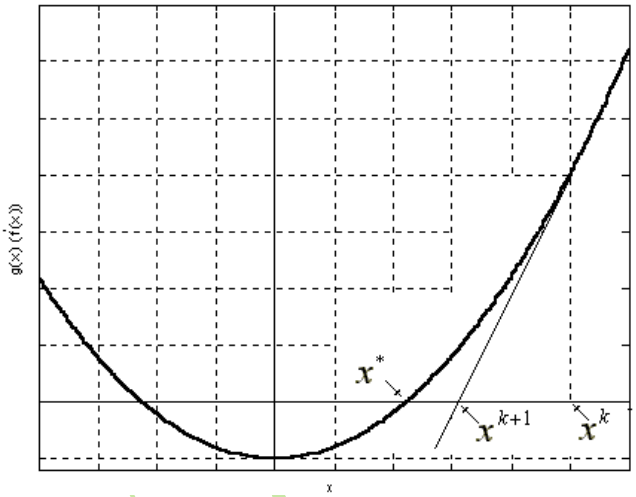
[۲۷] Scanning and Bracketing Procedures
[۲۸] Logical
[۲۹] Sequential
[۳۰] Convergence Rate
[۳۱] Super-linear
[۳۲] Indirect Methods

نکته – از دیدگاه آنالیز (محاسبات)عددی، مسئله محاسبه مینیمم f ، تقلیل پیدا می‌کند به حل غیرخطّی و محاسبه ریشه تابع $g(x)$ به‌طوریکه $g(x)$ مشتق مرتبه اوّل تابع هدف ($f(x)$) می‌باشد: $g(x)\ (\equiv f'(x)) = 0$

لذا، یک تقسیم‌بندی رایج نیز برحسب حدس اولیه به شکل زیر می‌باشد:
روش‌های تک‌نقطه‌ای[۳۳] – برای شروع تکرار و همچنین جمله عمومی تکرار، فقط به یک نقطه احتیاج است.
روش‌های دونقطه‌ای[۳۴] – برای شروع تکرار و همچنین جمله عمومی تکرار، به دو نقطه احتیاج است.
به‌طور مثال، همه روش‌های با تضمین همگرایی از این نوع هستند، چون ریشه را باید محاصره کنند.
نکته – بیش از ۲۰ روش برای حلّ عددی معادلات غیرخطی وجود دارد.

روش نیوتن (– رفسون) برای حل معادله غیرخطی $g(x)=0$

این روش از نوع تک‌نقطه‌ای است و ایده اصلی آن تقریب تابع $g(x)$ با یک خط راست است. این روش از اطلاعات هم متغیر مستقل و هم متغیر تابع (خود تابع $g(x)$ و مشتق آن، یا مشتق اول تابع هدف و مشتق دوم آن) استفاده می‌کند.
برای به‌دست آوردن جمله عمومی تکرار (موتور تکرار)، به شکل زیر توجه کنید. ایده گرافیکی روشن است، فرض کنید در حدس جاری (k ام) هستیم، یعنی x^k که فعلا آخرین مقدار نسبتاً دقیقی است که از اکسترمم (x^*) داریم. اگر رفتار تابع $g(x)$ را در این نقطه با یک خط که از x^k میگذرد و دارای شیبی برابر با مقدار مشتق $g(x)$ در x^k دارد، تقریب بزنیم، آنگاه محل تقاطع این خط با محور افقی (محور x ها) یک تقریبی از ریشه $g(x)=0$ می‌باشد.



با کمی بازآرایی و بازترکیب روابط به فرمول محاسبه خودکار x^{k+1} از روی اطلاعات x^k ، $g(x^k)$ و $g(x^{k+1})$ می‌رسیم:

$$x^{k+1} = x^k - \frac{g(x^k)}{g'(x^k)}$$

یا برحسب تابع هدف اصلی:

$$x^{nxt.} = x^{prv.} - \frac{f'(x^{prv.})}{f''(x^{prv.})}$$

به‌طوریکه بالانویس‌های $prv.$ و $nxt.$ به‌ترتیب به مفهوم "قبلی" و "بعدی" می‌باشند.
نکته – عبارت جمله عمومی تکرار را می‌توان به‌شکل (آموزنده)دیگری نیز به‌دست آورد. فرض کنید تابع هدف در فاصله حدود حدس اولیه تا جواب نهایی، یک رفتار درجه دوم داشته باشد، یعنی
بسط تیلور حول نقطه x^k :

$$f(x) = f(x^k) + f'(x^k)(x-x^k) + \frac{1}{2} f''(x^k)(x-x^k)^2$$

برای محاسبه اکسترمم می‌دانیم: $f'(x^*) = 0$
از طرفین رابطه بسط تیلور، مشتق نسبت به x می‌گیریم:

$$f'(x) = 0 + f'(x^k)\times 1 + (\frac{1}{2})(2) f''(x^k)(x-x^k) = 0 \ \rightarrow \ x^* = x^k - \frac{f'(x^k)}{f''(x^k)}$$

---
[۳۳] One-point formula
[۳۴] Two-point formula

مزایای روش نیوتن (- رفسون):

۱- مادامی که $f''(x) \neq 0$ ، آنگاه روش دارای سرعت همگرایی درجه دوم (کوادراتیک) می‌باشد.

۲- اگر تابع به‌طور محلی، رفتار درجه دوم داشته باشد، در یک حدس به جواب می‌رسیم.

معایب روش نیوتن (- رفسون):

۱. باید مشتق اول و دوم تابع هدف را به‌طور تحلیلی محاسبه کرد.

۲. اگر نزدیک نقطه عطف باشیم (یعنی $f''(x) \to 0$)، آنگاه روش بسیار کند می‌شود.

۳. احتمال چرخه[۳۵] و واگرایی[۳۶] وجود دارد.

اثبات این قضایا و نتایج و همچنین مثال‌های مربوطه در مراجع مربوطه (روش‌های محاسبات عددی) آمده‌است و لذا در اینجا تکرار نمی‌کنیم.

روش شبه نیوتن[۳۷]

ایده همان ایده نیوتن است، با این تفاوت که f' و f'' را نخواهیم تحلیلی به‌دست آوریم و مشتق‌ها را با اپراتورهای تفاضل می‌نویسیم. به‌همین خاطر ، مسمّا به روشهای "گام متغیر[۳۸]" هستند. به‌طور مثال اگر از تفاضل میانی[۳۹] استفاده کنیم، عبارت جمله عمومی تکرار به‌شکل زیر در می‌آید.

$$x^{k+1} = x^k - \frac{\left[f(x+h) - f(x-h)\right] / 2h}{\left[f(x+h) - 2f(x) + f(x-h)\right] / h^2}$$

نکته ۱ – می‌توان از اپراتورهای دیگر نیز استفاده کرد، نظیر اپراتور پیشرو[۴۰].

نکته ۲ – بحث راجع به پایداری و سرعت همگرایی در مراجع محاسبات عددی آمده‌است.

نکته ۳ – یک عیب این روش، محاسبات (فراخوانی تابع[۴۱]) بیشتر نسبت به روش نیوتن می‌باشد.

روش وتری[۴۲]

یکی از روش‌های دونقطه‌ای با تضمین همگرایی برای حل مسئله $g(x) = 0$ می‌باشد. فرض کنید دو نقطه x^p و x^q ، جواب مسئله (اکسترمم تابع هدف $f(x)$ یا ریشه $g(x) = 0$) را محاصره کرده‌باشند:

$$g(x^p) \times g(x^q) \prec 0 \quad \text{به‌طوریکه} \quad x^p \prec x^* \prec x^q$$

آنگاه نقطه بعدی (حدس ریشه) می‌تواند در وسط راه باشد: $\tilde{x}^* = \frac{x^p + x^q}{2}$

به‌طوریکه $\tilde{x}^*$ یک حدسی از x^* می‌باشد. اگر از جمله عمومی تکرار فوق‌المذکور استفاده کنیم، عملا از روش "تنصیف[۴۳]" استفاده کرده‌ایم. لازم به‌ذکر است که می‌توان ثابت کرد این روش معادل یکی از روش‌های جستجوی مستقیم نظیر تقسیم طلایی و یا فیبوناچی می‌باشد.

حال اگر کمی هوشمندی به‌خرج داده و از اطلاعات مقادیر تابع یعنی $g(x^p)$ و $g(x^q)$ نیز استفاده کرده و بگوییم حدس بعدی (تخمین x^*) محل تقاطع وتر زیر است:

---

[۳۵] Cycling

[۳۶] Divergence

[۳۷] Quasi-Newton

[۳۸] Variable Metrics Method

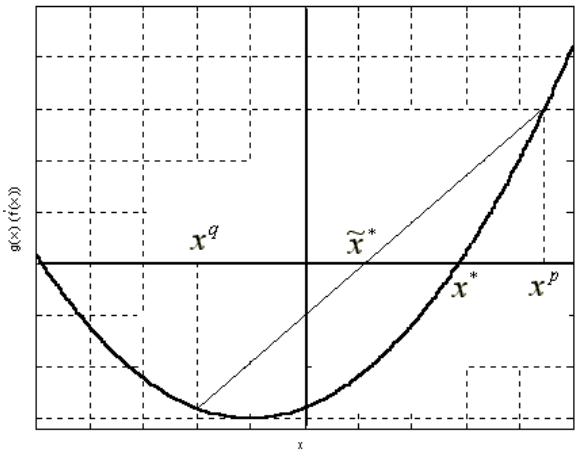
[۳۹] Central Difference

[۴۰] Forward Difference

[۴۱] Function Recall

[۴۲] Secant Method

[۴۳] Interval Halving - Bisection



معادله خطی که از دو نقطه $p:[x^p, g(x^p)]$ و $q:[x^q, g(x^q)]$ می‌گذرد، عبارتست از:

$$\frac{y-g(x^p)}{g(x^q)-g(x^p)}=\frac{x-x^p}{x^q-x^p}$$

آنگاه محل تقاطع وتر (خط مزبور) با محور افقی:

$$\frac{0-g(x^p)}{g(x^q)-g(x^p)}=\frac{\tilde{x}^*-x^p}{x^q-x^p} \quad \rightarrow$$

$$\tilde{x}^*=x^p-(x^q-x^p)\times\frac{g(x^p)}{g(x^q)-g(x^p)}$$

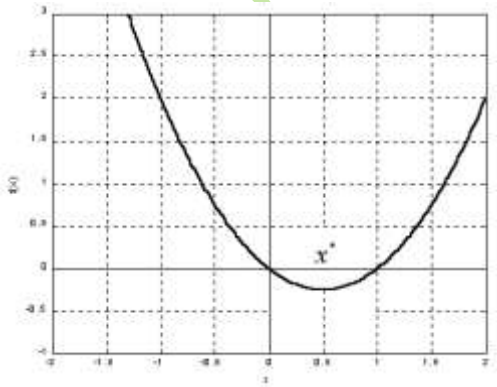
و اگر در مقام مشابهت[۴۴] با نیوتن یا شبه‌نیوتن بنویسیم:

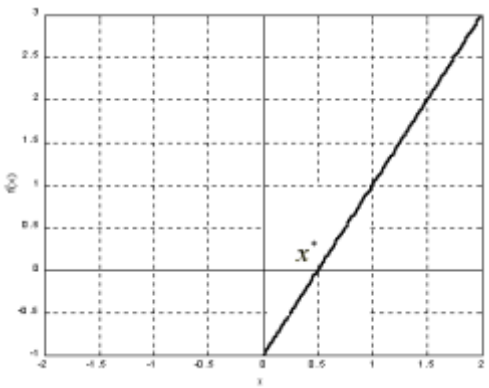
$$\tilde{x}^*=x^p-\frac{g(x^p)}{[g(x^q)-g(x^p)]/(x^q-x^p)}$$

نکات مربوط به روش وتری:

۱. در مقام مشابهت، این روش نیز نوعی نیوتن یا شبه‌نیوتن می‌باشد.
۲. محاسبه $g(x^p)$ و $g(x^q)$ می‌تواند به‌صورت اپراتور تفاضل صورت بگیرد.
۳. می‌توان ثابت کرد که این روش (روش وتری) نوعی روش جستجوی مستقیم است.
۴. می‌توان ثابت کرد که سرعت همگرایی آن معادل $\frac{1+\sqrt{5}}{2}$ (یا تقریبا 1.6) است.
۵. اسامی دیگر این روش، رگولا- فالسی[۴۵] یا معادلاً به انگلیسی False Position می‌باشد.
۶. اگر اصرار نداشته‌باشیم که ریشه محاصره شود، اولا)همگرایی تضمین‌شده نیست، ثانیا) با یک گروه دیگر از روش‌های دونقطه‌ای مثل وگشتاین[۴۶] یا شبه‌نیوتن روبرو خواهیم بود.
۷. دقّت شود بعد از محاسبه $\tilde{x}^*$، باید برای ادامه یکی از نقاط x^p و x^q را برای حفظ دونقطه انتخاب کنیم. اگر همگرایی تضمینی بخواهیم، معیار انتخاب x^p یا x^q آنست که مقدار تابع $g(\tilde{x}^*)$ با تابع $g(x^p)$ یا $g(x^q)$ مختلف‌العلامه باشد. و اگر اصرار به همگرایی نیست، همیشه دو نقطه اخیر را به‌کار می‌گیریم.

مثال۶-۷. مقایسه سه روش غیرمستقیم گفته‌شده برای تابع مجذوری (کوادراتیک): $f(x)=x^2-x$





[۴۴] Analogy
[۴۵] *Regula Falsi*
[۴۶] Wegestein

حلّ: چون می‌خواهیم حتما به جواب برسیم، باید ابتدائا بازه اولیه را افزون بر فرض یونیمودال مشخص کنیم. یک راه‌حل این است که از کوچکترین مقدار ممکن شروع کنیم (مثلا -10^{16} یا یک حدس فیزیکی معقول) و سپس مرتبا با گام‌های نسبتا کوچک جلوتر بیایم تا ببینیم آیا مقدار تابع کوچک می‌شود یا خیر. سپس با یک سیاستی گام‌ها را بزرگتر کرده تا ببینیم f دارد بزرگ می‌شود، آن موقع جواب را محاصره کرده‌ایم. لازم به‌ذکر است، این درست مثل این‌است که با $g(x)=f'(x)$ کارکنیم، ببینیم کی علامت عوض می‌شود. به‌هرحال برای این مثال بازه $x\in[-3,3]$ را انتخاب کرده‌ایم.

← روش نیوتن: چون این روش یک‌نقطه‌ای است، به یک حدس اولیه نیاز داریم:

$$\begin{cases}\dot{x}=3\\ f'(x)=2x-1\\ f''(x)=2\end{cases}\quad\rightarrow\quad x^1=x^0-\frac{f'(x^0)}{f''(x^0)}=3-\frac{5}{2}=0.3$$

دقت شود فقط در یک تکرار به جواب رسیده‌ایم. علت این است که تابع هدف کوادراتیک است و لذا مشتق آن یک خط راست است.

← روش شبه‌نیوتن: از اپراتور تفاضل پیشرو برای مشتق اول و تفاضل میانی سه‌نقطه‌ای برای مشتق دوم استفاده می‌کنیم، طول گام را نیز $h=0.001$ انتخاب می‌کنیم:

$$x^{k+1}=x^k-\frac{\left[f(x+h)-f(x)\right]/h}{\left[f(x+h)-2f(x)+f(x-h)\right]/h^2}$$

اگر تولرانس توقف 10^{-6}باشد، آنگاه یک سعی دیگر نیاز است.

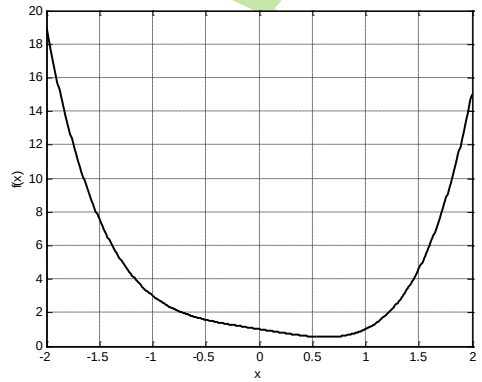
← روش وتری: این روش دونقطه برای شروع نیاز دارد:

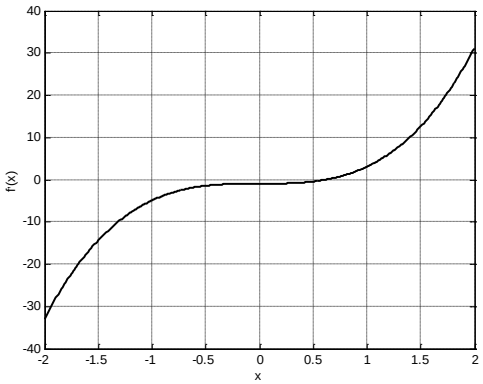
$$x^p=-3\,,\,x^q=3\rightarrow f'(-3)=-7\,,\,f'(3)=5$$

$$x^1=-3-(3+3)\frac{-7}{5-(-7)}=-3+3.5=0.5$$

چون تابع هدف کوادراتیک است (مشتق یک خط راست می‌باشد) بازهم در یک سعی به جواب رسیدیم.

مثال۶- مقایسه سه روش گفته‌شده برای یک مسئله سخت‌تر: $f(x)=x^4-x+1$ (دقت کنید که تابع پهن است و تیز نیست)





مطلوبست محاسبه مینیمم با تولرانس 10^{-7}، برای روش شبه‌نیوتن از $h=0.1$ استفاده کنید و برای روش وتری، دو نقطه اوّل را -3 و 3 فرض کنید.

← روش نیوتن (- رفسون)

$$x^1 = x^0 - \frac{f'(x^0)}{f''(x^0)} = 2.009259$$

$$x^2 = x^1 - \frac{f'(x^1)}{f''(x^1)} = 1.36015$$

$$\vdots$$

$$x^8 = x^7 - \frac{f'(x^7)}{f''(x^7)} = 0.6299605$$

$$x^9 = x^8 - \frac{f'(x^8)}{f''(x^8)} = 0.6299605$$

← روش شبه‌نیوتن: از فرمول تفاضل میانی

$$x^{k+1} = x^k - \frac{h}{2}\frac{f(x^k+h) - f(x^k-h)}{f(x^k+h) - 2f(x^k) + f(x^k-h)}$$

k	$\tilde{x}^*$
0	3.0000
1	2.00833
2	1.35816
⋮	⋮
8	0.624669
9	0.624669313

تعداد تکرار = ۹

← روش وتری:

$$\begin{cases} \tilde{x}^* = x^q - \dfrac{f'(x^q)}{\left[f'(x^q) - f'(x^p)\right] \big/ (x^q - x^p)} \\ g(x) = f'(x) = 0 \end{cases}$$

k	x^q	x^p	$g(x^p)$ or $f'(x^p)$
0	3.0	–3.0	–109.0
1	3.0	0.027778	–0.9991
2	3.0	0.055296	–0.9992
⋮	⋮	⋮	⋮
100	3.0	0.6299311	-1.399×10^{-4}
⋮	⋮	⋮	⋮
132	3.0	0.6299597	-3.952×10^{-6}

تعداد تکرار = ۱۳۲

روش‌های مبتنی بر برازش منحنی[۴۷]

کلاس دیگری از روش‌های جستجو روش‌های مبتنی بر تقریب چند جمله‌ای می‌باشد و معمولاً یا درجه دوم[۴۸] (معروف به روش پاول[۴۹]) یا درجه سوم[۵۰] (معروف به روش دیویدن[۵۱]) هستند.

میان‌یابی (برون‌یابی) درجه دوم

می‌دانیم که از $n+1$ زوج نقطه یک چندجمله‌ای درجه n می‌گذرد، لذا اساس میان‌یابی یا برازش (دقیق) منحنی درجه دوم استفاده از سه نقطه است ولی ما در روش‌های الگوریتمیک معمولاً با یک نقطه شروع می‌کنیم (حدس اولیه)، لذا بحث را به دو بخش متمرکز می‌کنیم، یکی اساس ایده و دیگری نحوه محاصره مینیمم، چون در عمل با بازه و گستره‌های باز روبرو هستیم.

اساس ایده – از سه نقطه، یک منحنی درجه دوم می‌گذرد، پس فرض می‌کنیم سه نقطه x_1، x_2 و x_3 به‌ترتیب صعودی داریم، یعنی $x_1 \prec x_2 \prec x_3$ (نامساوی است و اصلا مساوی نداریم) همچنین مقادیر تابع هدف را در این سه نقطه داریم، یعنی

$$f_1 \equiv f(x_1),\ f_2 \equiv f(x_2),\ f_3 \equiv f(x_3)$$

و مقدار مینیمم حتما در این فاصله است (بین دو حالت حدی $x_1 \prec x^* \prec x_3$)

اگر تابع هدف در این بازه رفتار درجه دوم داشته باشد، یعنی

$$f(x) \approx \phi(x) = a + bx + cx^2$$

کافیست با استفاده از سه نقطه بالا، a، b و c را به‌دست آورده و از روی $\phi(x)$ مقدار مینیمم را با تقریب به‌دست آوریم:

$$\phi'(x) = b + 2cx = 0 \quad \rightarrow \quad \tilde{x}^* = -\frac{b}{2c}$$

نحوه به‌دست آوردن ضرایب a، b و c:

$$\begin{cases} f_1 = a + bx_1 + cx_1^2 \\ f_2 = a + bx_2 + cx_2^2 \\ f_3 = a + bx_3 + cx_3^2 \end{cases} \rightarrow c \text{ و } b \text{ ،} a \text{ محاسبه} \qquad \rightarrow \tilde{x}^* \text{ محاسبه}$$

$$\tilde{x}^* = \frac{1}{2}\left\{\frac{\left(x_2^2 - x_3^2\right)f_1 + \left(x_3^2 - x_1^2\right)f_2 + \left(x_1^2 - x_2^2\right)f_3}{\left(x_2 - x_3\right)f_1 + \left(x_3 - x_1\right)f_2 + \left(x_1 - x_2\right)f_3}\right\}$$

برای حالتی که x_1، x_2 و x_3 یا f_1، f_2 و f_3 خیلی به‌هم نزدیک باشند، این فرمول بهتر است:

$$\tilde{x}^* = \frac{1}{2}\left(x_1 + x_2\right) + \frac{1}{2}\left\{\frac{\left(x_2 - x_3\right)\left(x_3 - x_1\right)\left(f_2 - f_1\right)}{\left(x_2 - x_3\right)f_1 + \left(x_3 - x_1\right)f_2 + \left(x_1 - x_2\right)f_3}\right\}$$

بنابراین چهارنقطه (بازیکن) در این صحنه (نمایش) داریم: $\tilde{x}^*$، x_1، x_2، x_3 و مقادیر تابع متناظرشان. برای زنده‌بودن و ادامه الگوریتم – که بالذات تکراریست- باید سه نقطه از چهار نقطه را انتخاب کنیم.

روش و روح کار، حذف آن نقطه‌ای است که بیشترین مقدار را دارد، به‌شرطیکه همچنان نقطه بهینه را محاصره کرده باشیم. چون سه نقطه داریم (دو تا بازه داریم)، پس $\tilde{x}^*$ دو حالت دارد، یا بین x_2 و x_1 (بازه اول) است یا بین x_3 و x_2 (بازه دوم) و چون در هر بازه ممکن است مقدارش کم یا زیادتر از نقطه رقیب باشد، چهار حالت واقعی (فرض یونیمودال) پیش می‌آید وگرنه شش حالت داریم:

$$x_1 \prec \tilde{x}^* \prec x_2 \qquad f^* \prec f_2,\ f^* \prec f_1 \qquad f^* \succ f_2,\ f^* \prec f_1 \qquad f^* \succ f_2,\ f^* \succ f_1$$

---

[۴۷] Polynomial Approximation Methods

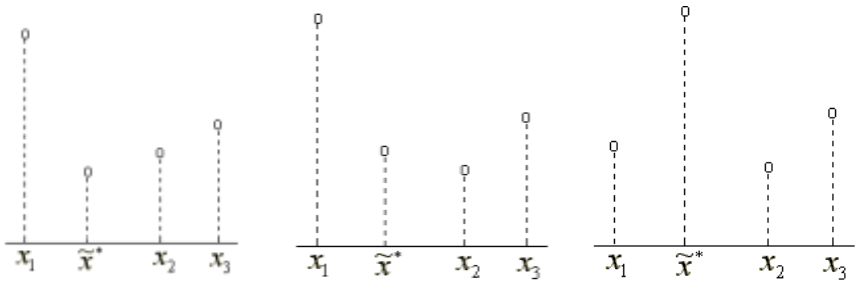
[۴۸] Quadratic

[۴۹] Powell Method

[۵۰] Cubic Interpolation

[۵۱] Davidon Method

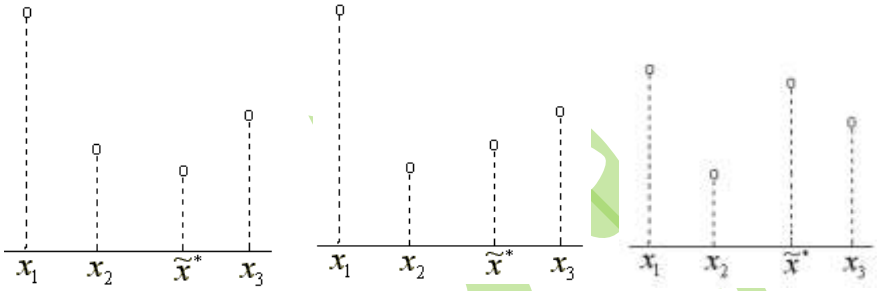
بازه اول، پس با f_1 و f_2 محک می‌زنیم.



Pick $(x_1, \tilde{x}^*, x_3)$ | Pick $(\tilde{x}^*, x_2, x_3)$ | Numerical Error

$x_2 \prec \tilde{x}^* \prec x_3$ | $f^* \prec f_2$, $f^* \prec f_3$ | $f^* \succ f_2$, $f^* \prec f_3$ | $f^* \succ f_2$, $f^* \succ f_3$

بازه دوم، پس با f_3 و f_2 محک میزنیم.



Pick $(x_2, \tilde{x}^*, x_3)$ | Pick $(x_1, x_2, \tilde{x}^*)$ | Numerical Error

نکته: اگر به حالت Numerical Error برخوردید، چند کار می‌توانید بکنید. یا به‌طور رندام سه نقطه دیگر را انتخاب کنید، یا به مینیمم محلی برخورده‌اید و فرض یونیمودال برقرار نیست، پس برگردید سه نقطه دیگر انتخاب کنید یا از کاربر کمک بخواهید.

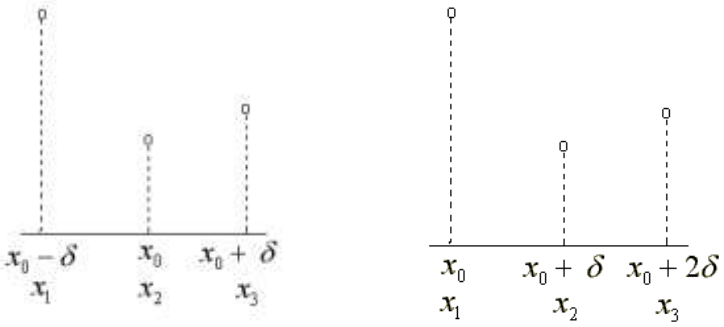
نکته: روش میان‌یابی گفته‌شده را می‌توان با دو نقطه و یک مشتق انجام داد و اگر مشتق تحلیلی سخت است می‌توان مشتق عددی گرفت ولی متداول استفاده از سه مقدار تابع می‌باشد.

نکته: در روش کامپیوتری، متداول این است که یک حدس اولیه در اختیار داریم، در حالی‌که این روش بر مبنای سه نقطه کار می‌کند. یک روش پیشنهادی (روش پاول) برای مینیمم‌سازی به شرح زیر است:

الف: علاوه بر حدس اوّلیه، از کاربر یک طول گام (مثلا δ) بپرسید.

ب : مقدار $f(x_0)$ و $f(x_0+\delta)$ را محاسبه کنید.

ج : اگر $f(x_0) \prec f(x_0+\delta)$ باشد، آنگاه سومین نقطه را $x_0-\delta$ در نظربگیرید. در غیراین‌صورت (یعنی $f(x_0) \succ f(x_0+\delta)$) ، آنگاه سومین نقطه را $x_0+2\delta$ درنظربگیرید.



نکته : به‌طور تجربی اثبات شده که میان‌یابی درجه دوم بهتر از تقسیم طلایی کار می‌کند.

میان‌یابی درجه سوم

این روش نیز مانند روش قبلی، تابع هدف را با یک چندجمله‌ای تقریب می‌زند. درجه چندجمله‌ای، درجه سوم است.

$$f(x) \approx \phi(x) = a_1x^3 + a_2x^2 + a_3x + a_4$$

می‌توان هم با چهارنقطه و مقدار تابع کار کرد و هم با دو نقطه، مقادیر تابع و مشتق تابع هدف. اگر از روش اول استفاده کنیم – با استفاده از (x_1, f_1)، (x_2, f_2)، (x_3, f_3) و (x_4, f_4) – برای محاسبه ضرایب a_1، a_2، a_3 و a_4 باید دستگاه جبری زیر را حل کنیم:

$$\begin{cases} a_1x_1^3 + a_2x_1^2 + a_3x_1 + a_4 = f_1 \\ a_1x_2^3 + a_2x_2^2 + a_3x_2 + a_4 = f_2 \\ a_1x_3^3 + a_2x_3^2 + a_3x_3 + a_4 = f_3 \\ a_1x_4^3 + a_2x_4^2 + a_3x_4 + a_4 = f_4 \end{cases}$$

محاسبه اپتیمم:

$$\phi'(x) = 3a_1x^2 + 2a_2x + a_3 = 0 \quad \rightarrow \tilde{x}^* = \frac{-2a_2 \pm \sqrt{4a_2^2 - 12a_1a_3}}{6a_1}$$

یکی از ریشه‌ها، مربوط به مینیمم و دیگری مربوط به ماگزیمم است، لذا برای انتخاب باید علامت مشتق ثانی را در نظر گرفت.

$$\phi''(x) = 6a_1x + 2a_2 \quad \rightarrow \quad sign(6a_1\tilde{x}^* + 2a_2): \begin{cases} if \prec 0, & minimum \\ if \succ 0, & maximum \end{cases}$$

و اگر از روش دوم استفاده کنیم – استفاده از (x_1, f_1)، (x_1, f_1')، (x_2, f_2) و (x_2, f_2') – برای محاسبه ضرایب a_1 ، a_2 ، a_3 و a_4 باید دستگاه جبری زیر حل کنیم:

$$\begin{cases} a_1x_1^3 + a_2x_1^2 + a_3x_1 + a_4 = f_1 \\ 3a_1x_1^2 + 2a_2x_1 + a_3 = f_1' \\ a_1x_2^3 + a_2x_2^2 + a_3x_2 + a_4 = f_2 \\ 3a_1x_2^2 + 2a_2x_2 + a_3 = f_2' \end{cases}$$

می‌توان ثابت کرد $\tilde{x}^*$ از رابطه تحلیلی زیر به‌دست می‌آید:

$$\tilde{x}^* = x_2 - \left\{ \frac{f_2' + w - z}{f_2' - f_1' + 2w} \right\}(x_2 - x_1), \; z \equiv 3\frac{f_1 - f_2}{x_2 - x_1} + f_1' + f_2', \; w \equiv \sqrt{z^2 - f_1'f_2'}$$

به‌شرطی‌که x_1 و x_2 ، مینیمم را محاصره کرده باشند، یعنی $x_1 \prec x_2 \quad , \quad f_1' \prec 0 \quad , \quad f_2' \succ 0$

برای تکرار میان‌یابی تا حصول مینیمم با دقّت کافی، دوباره همان مباحث را برای میان‌یابی درجه دوم داریم.

روش هیبرید برنت[۵۲]

روش جستجوی برنت یک روش جستجوی خطی می‌باشد که عملا ترکیبی از جستجوی تقسیم طلایی و میان‌یابی درجه دوم می‌باشد[1]. همان‌طور که می‌دانیم جستجوی تقسیم طلایی دارای سرعت همگرایی درجه اول و در طرف دیگر روش‌های میان‌یابی (چندجمله‌ای) دارای سرعت همگرایی مجانبی (تندتر از سرعت ابرخطی) می‌باشند. مشخصه همگرایی جستجوی تقسیم طلایی (درجه اول) از همان ابتدای تکرار برقرارست، درحالی‌که روش‌های میان‌یابی علیرغم سرعت همگرایی خوب، معمولا در انتهای جستجو از خود نوسان نشان می‌دهند، لذا برنت یک روش ترکیبی جستجو ابداع کرد که در عین حفظ مزایای هر دو روش از معایب آنها بپرهیزد.

در این روش، باید ابتدائا از همان روتین تقسیم طلایی برای تقسیم بازه‌های تردید استفاده کنیم. افزون براین چند نقطه دیگر هم محاسبه می‌شوند تا برای میان‌یابی و تعیین مینیمم به‌کار گرفته شوند. اگر مینیمم به‌دست‌آمده در بازه تردید مناسب قرار داشت، آنگاه یک مرحله میان‌یابی دیگر را پیش می‌بریم. در غیراین‌صورت (خارج بازه تردید) یک گام جستجوی دیگر ولی با روش تقسیم طلایی را پیش می‌بریم.

[۵۲] Brent's Hybrid Method

روش هیبرید تنصیف- میان‌یابی درجه سوم

این روش ترکیبی از دو الگوریتم تنصیف و میان‌یابی درجه سوم بهره می‌گیرد[2]. ابتدای الگوریتم با تنصیف شروع می‌شود، به‌طوریکه یک نقطه در میانه بازه تردید انتخاب شده و مقدار تابع به همراه مشتق آن حساب می‌شود. بدین ترتیب (برای تنصیف خالص) نصف بازه تردید از دامنه جستجو حذف می‌شود. در روش هیبریدی از نقاط موجود، یک میان‌یابی درجه سوم انجام شده و نقطه مینیمم حساب می‌شود. سپس چک می‌شود آیا مینیمم محاسبه‌شده خارج از بازه تردیدست یا داخل آن. در صورتی‌که داخل بازه قرار بگیرد، آنگاه حذف یا کاهش بازه تردید را با روش میان‌یابی ادامه می‌دهیم. در غیر این صورت (خارج بازه تردید) مسئله را با تنصیف ادامه می‌دهیم.

روش هیبرید کارالامبوس[۵۳]

این روش نیز یک روش ترکیبی (وتری – میان‌یابی) است که سرعت بیشتری دارد [3]. برای جزئیات آن به مرجع مربوطه باید مراجعه کرد.

روش هیبرید میان‌یابی یا عقب‌گرد[۵۴]

این روش نیز یک روش ترکیبی (میان‌یابی درجه دوم و سوم) است که از سرعت خوبی به‌ویژه در روش‌های بهینه‌سازی شبه- نیوتنی برخوردارست [4]. ایده اساسی بر انتخاب گام مناسب تقریب جبری مشتق برای میان‌یابی می‌باشد. ابتدا یک گام اولیه در نظر گرفته می‌شود، سپس دو میان‌یابی درجه دوم و سوم انجام شده و بر اساس آنها مینیمم تابع هدف محاسبه می‌شود. سپس با یک شاخص مبنا این دو نقطه بهینه مقایسه می‌شوند و در صورت عدم ارضا، گام مزبور کاهش داده می‌شود(عقب‌گرد) و محاسبات مجدداً تکرار می‌شود. شرط اختتام، عدم مغایرت (با یک تولرانس از پیش معلوم) دو تقریب به‌دست‌آمده از نقطه بهینه می‌باشد. برای جزئیات بیشتر باید به مرجع مربوطه مراجعه کرد.

مراجع

[1]. Brent, R.P. *Algorithms for Minimization Without Derivatives*, Englewood Cliffs, N.J., Prentice-Hall, 1973.

[2]. Scales, L.E. *Introduction to Non-Linear Optimization*, New York: Springer-Verlag, 1985.

[3] Charalambous, C., “Conjugate Gradient Algorithm for Efficient Training of Artificial Neural Networks “, IEEE Proceedings, Vol. 139, No. 3, pp. 301-310, 1992.

[4]. Dennis, J.E. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Englewood Cliffs, N.J., Prentice-Hall, 1983.

[۵۳] Charalambous

[۵۴] Back-Tracking